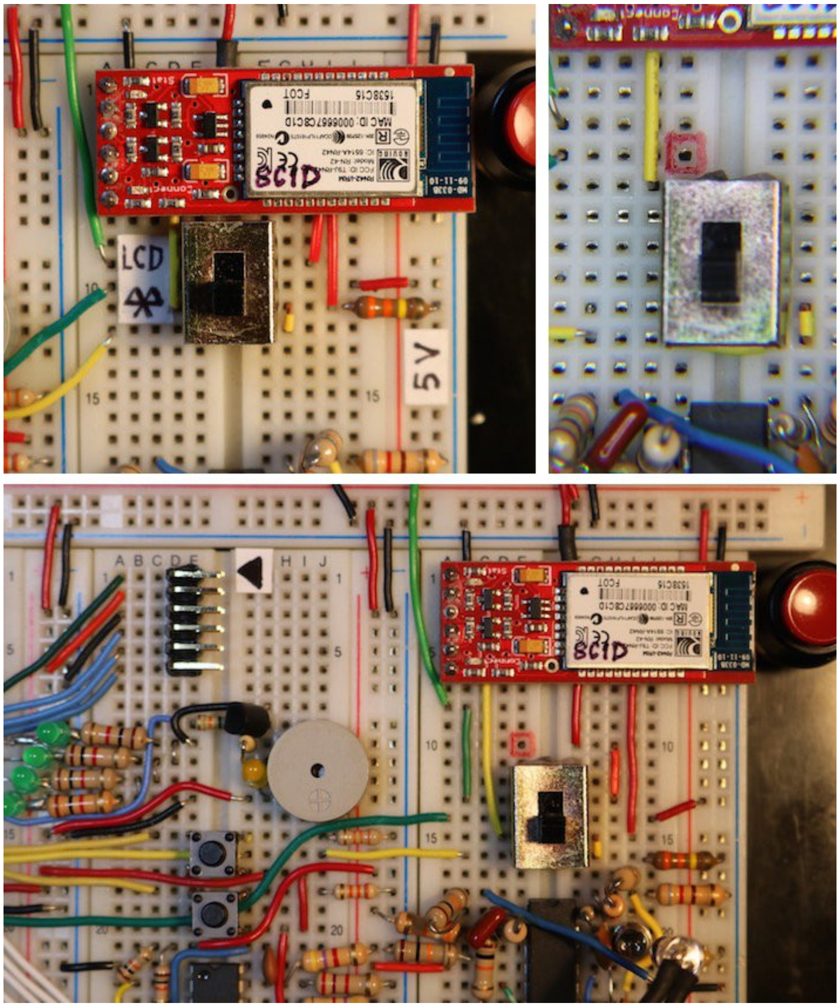
Bug Report

BME 363 Biomedical Instrumentation Design Laboratory, University of Rhode Island

1. Breadboard Connection Problem

Problem

This happened on the very first demo breadboard that had been working for a couple of years. The LCD display and the Bluetooth modem suddenly stopped working. Both could be powered up but failed to show and data.



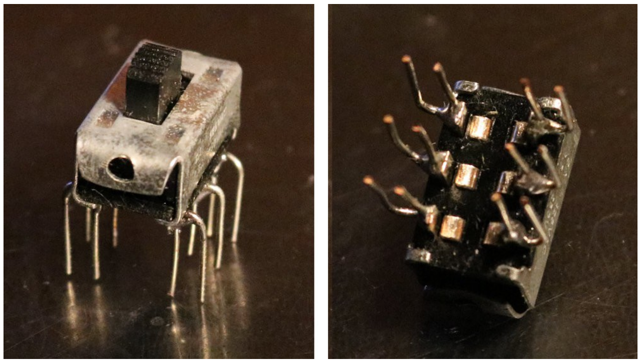
Debugging

Visual inspection showed that all wires were properly connected. Connectivity tests using a multimeter showed that the Tx line from the PIC (pin 25) did not connect to either the Rx line of the LCD or the Bluetooth modem. The DPDT switch was removed and checked; no bad connection within the switch was found. So the problem was narrowed down to the hole on the breadboard where the switch was supposed to make contact with the Tx line. Visual inspection of the hole (marked by red frame) revealed that the clips under the hole were stretched open, likely due to repetitively motions of the switching.

Solution

The DPDT switch was moved to a nearby position avoiding the defective hole. After rewiring, the breadboard was functional again.

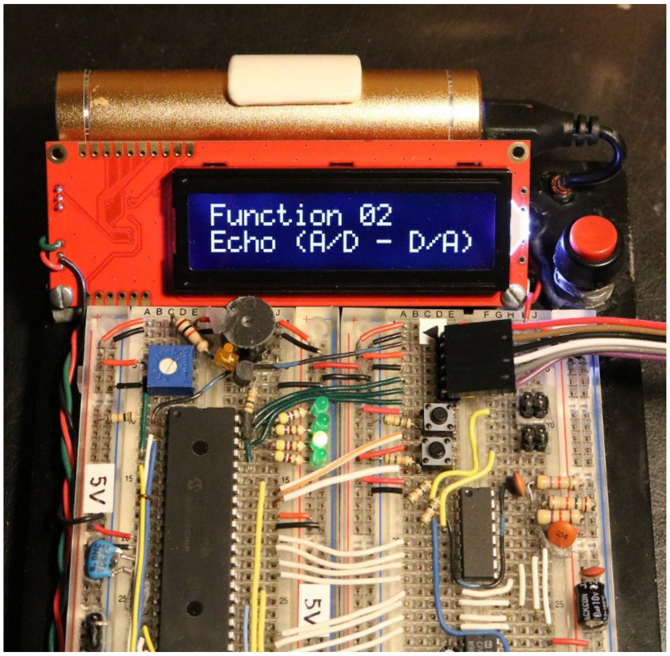
The cause of this problem is the lateral motions of sliding the switch, which results in loose contact of the clippers inside the breadboard. A more permanent solution is to solder extra legs to the switch as shown. In addition to the 6 legs under the DPDT switch, 6 lateral legs provide better connections to the breadboard.

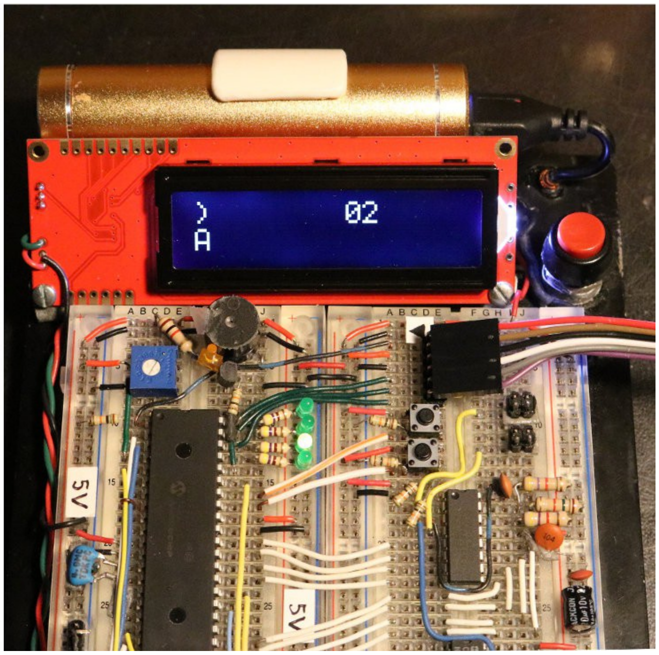


2. LCD Display

Problem

We have seen this problem several times. The LCD display suddenly stopped working properly. The correct display is shown on the left. The malfunctioning display shows the function number correctly, but does not show "Function" and the description of the second line.





Debugging

A careful examination reveals that the characters are actually printed on the screen but the cursor position is not advanced. The cursor is supposed to advance to the next position after each printing. The original code for string printing is done by calling SetPosition() first following by calling PrintLine(), which is shown below:

```
void PrintLine(const unsigned char *string, unsigned char numChars){
   unsigned char count;
   for (count=0; count<numChars; count++) Transmit(string[count]);
}
```

Solution

The problem is resolved by advancing the cursor explicitly with the new PrintLineAt():

```
void PrintLineAt(const unsigned char *string, unsigned char numChars, unsigned char position){
   unsigned char count;
   for (count=0; count<numChars; count++) {
      SetPosition(position);
      Transmit(string[count]);
      position++;
   }
}
```

3. Noise in PPG Caused by Waveform Generation

Problem

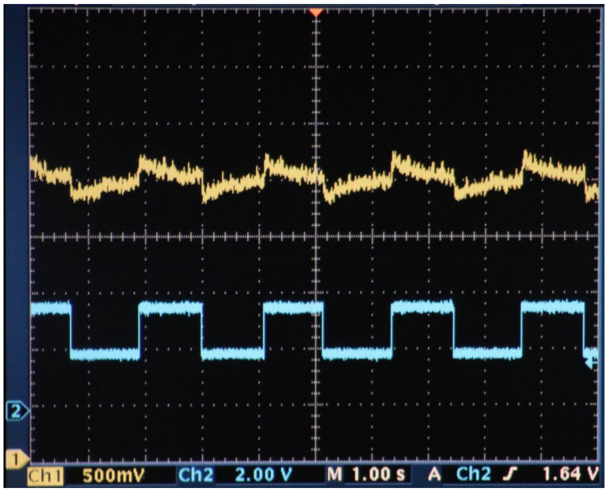
There was an undesirable cross-talk between the PPG circuit and the waveform generation circuit. The fast-switching square wave from the LM556 somehow affected the PPG waveform. In the top figure, channel 1 (yellow) shows the PPG without finger inserted waveform generator output (AN0, pin 2 of PIC). In the next figure, with finger inserted, the PPG waveform was affected at certain points such as the one shown by the red arrow. The problem could be accentuated for a small-magnitude PPG.

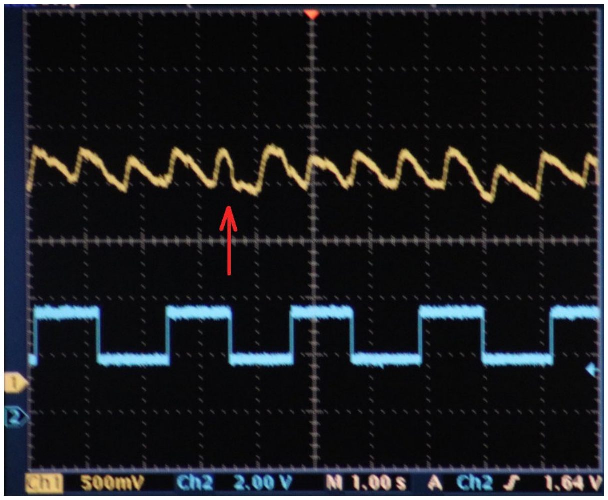
Debugging

The coupling could have come through the power supply. Adding a large capacitor (such as 220 μF) between the 5V line and the ground line could suppress but not completely eliminate the noise.

Solution

The noise went away when the waveform generator circuit (LM 556) was turned off. Thus, a switch was added; the Vcc and the two reset lines were connected to 5V via the switch. When the PPG is in use, the waveform generator can be manually turned off to get rid of the noise problem.


2
1
Ch1 500mV
Ch2 2.00 V
M 1.00 s
A Ch2 ∫ 1.64 V


1
2
Ch1 500mV
Ch2 2.00 V
M 1.00 s
A Ch2 ∫ 1.64 V